

Serial No. 09/816,940

Art Unit: 2178

REMARKS

Claims 1-23 are pending in the application. Claims 21-23 are newly added.

Claims 1-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Stone et al., UNIX Fault Management: A Guide for System Administration, Chapters 3-6 and 9, hereinafter "Stone". Applicant respectfully traverses this rejection.

Claim 1 provides a method carried out by a status engine for monitoring services of an information technology (IT) environment. The method is based on a service model including service model elements that each represent a service of the IT environment and are each associated with a service model status. The service model elements include at least one superordinate service model element and at least one subordinate service model element.

The method includes calculating a status of the at least one superordinate service model element by considering status dependency and propagation between the service model elements within the service model. The status of the at least one superordinate service model element depends on a status of the at least one subordinate service model element.

The method calculates the status of the at least one superordinate service model element according to one or more rules. The rules define the dependency of the status of the at least one superordinate service model element on the status of the at least one subordinate service model element and a propagation of the status from the at least one subordinate service model element to the at least one superordinate service model element.

The rules include at least one of a) a rule that is based on additional attributes of at least one of the service model elements other than the service model status, b) a rule

Serial No. 09/816,940

Art Unit: 2178

that ignores the at least one subordinate service model element, c) a rule that is defined by a user on the basis of at least one of i) logical and ii) arithmetical operations of the status or the attributes of the at least one subordinate service model element, and d) a rule that is programmed individually by a user.

Stone discloses tools and methods for monitoring events that can occur on UNIX systems. Stone discloses the generation of messages in response to events occurring at the IT services. An example of an event is the failure of a particular IT service. The generated messages transport the status of the event to a management station where they can be viewed by a user. The status of an IT service is assessed solely on the basis of messages reporting the severity of the event occurring at the corresponding IT service. The statuses of different services of the IT network are not interconnected and do not depend on each other.

Stone also discloses agents that "filter out redundant or unnecessary events before forwarding them to the management station" (page 18, lines 15-16). Stone filters events or combines several low-level events to one high-level event (e.g. several failures to log onto a system may create a high-level event of "unauthorized hacker"). This feature avoids the generation of messages transporting a status but does not interconnect the statuses of different services with each other. Thus, the filters described in Stone relate to events but not to statuses of IT services and their propagation and calculation.

As described above, the statuses of different services of the IT network are not interconnected and **do not depend** on each other in the management station disclosed in Stone. In contrast, claim 1 defines that the status calculation of a service in the service model **depends on** the status of other services in the service model. Thus, Stone does not disclose a method for calculating the status of at least one superordinate service model element "by considering **status dependency** and propagation between the service model elements within the service model," nor does Stone disclose a method "wherein the status of the at least one superordinate service

Serial No. 09/816,940

Art Unit: 2178

model element **depends on** a status of the at least one subordinate service model element". Furthermore, Stone does not disclose rules that "define the **dependency** of the status of the at least one superordinate service model element on the status of the at least one subordinate service model element and a propagation of the status from the at least one subordinate service model element to the at least one superordinate service model element."

Thus, Stone fails to disclose or suggest the elements of claim 1. Therefore, claim 1 is not anticipated by Stone.

Claims 2-8 and 21 depend from claim 1. For at least the above reasons, claims 2-8 and 21 are also patentable over Stone.

Independent claim 9 recites features similar to those recited in claim 1. For reasons similar to those described above, Stone does not disclose a computer system "wherein a status of the at least one superordinate service model element **depends on** a status of the at least one subordinate service model element," nor does Stone disclose a status engine that "can calculate the status of said superordinate service model element by considering **status dependency** and propagation between the service model elements within the service model," as recited in claim 9. Furthermore, Stone does not disclose rules that "define the **dependency** of the status of the at least one superordinate service model element on the status of the at least one subordinate service model element and a propagation of the status from the at least one subordinate service model element to the at least one superordinate service model element." Therefore, claim 9 is not anticipated by Stone.

Claims 10-15 and 22 depend from claim 9. For at least the above reasons, claims 10-15 and 22 are also patentable over Stone.

Independent claim 16 recites features similar to those recited in claim 1. For reasons similar to those described above, Stone does not disclose a computer program

Serial No. 09/816,940

Art Unit: 2178

product for carrying out a method for calculating the status of at least one superordinate service model element "by considering **status dependency** and propagation between the service model elements within the service model," nor does Stone disclose a computer program product "wherein the status of the at least one superordinate service model element **depends on** a status of the at least one subordinate service model element". Furthermore, Stone does not disclose rules that "define the **dependency** of the status of the at least one superordinate service model element on the status of the at least one subordinate service model element and a propagation of the status from the at least one subordinate service model element to the at least one superordinate service model element." Therefore, claim 16 is not anticipated by Stone.

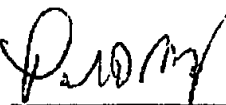
Claims 17-20 and 23 depend from claim 16. For at least the above reasons, claims 17-20 and 23 are also patentable over Stone.

For the reasons set forth above, the rejection of claims 1-23 under 35 U.S.C. 102(b) as anticipated by Stone is overcome. Applicant respectfully requests that the rejection of claims 1-20 be reconsidered and withdrawn.

An indication of the allowability of all pending claims by issuance of a Notice of Allowability is earnestly solicited.

Respectfully submitted,

Date: 12-2-04



Paul D. Greeley
Reg. No. 31,019
Attorney for Applicants
Ohlandt, Greeley, Ruggiero & Perle, LLP
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: (203) 327-4500
Fax: (203) 327-6401